

Квадрокоптер

Бондаренко Дмитро, Будніков Іван, Лазоренко Марина

КНУ імені Т.Г.Шевченка, фізичний факультет

December 25, 2015

Overview

- 1 Кватерніон
- 2 Матриця повороту для кватерніонів
- 3 Матриця повороту для кутів Ейлера
- 4 Переваги кватерніонів у визначенні поворотів
- 5 Карданнів підвіс, Gimble lock
- 6 Отримання кутів
- 7 Фільтр Мадвіка

Кватерніон — гіперкомплексне число, яке реалізується в 4-вимірному просторі. Вперше описане В. Р. Гамільтоном у 1843 році.

Кватерніон має вигляд $a + bi + cj + dk$, де

a, b, c, d , — дійсні числа;

i, j, k , — уявні одиниці, що задовольняють співвідношенням

$i^2 = j^2 = k^2 = ijk = -1$, з яких випливають ще й такі співвідношення:

$$\begin{aligned} ij &= -ji = k, \\ jk &= -kj = i, \\ ki &= -ik = j. \end{aligned}$$

Часто замість i, j, k використовують позначення для уявних одиниць відповідно i_1, i_2, i_3 , а також покладають $i_0 := 1$.

Ще один, зрідка вживаний, варіант позначень: e_0, e_1, e_2, e_3 .

Кватерніон

Кватерніон $q = a + bi + cj + dk$ можна представити у вигляді пари скаляра та 3-вимірного вектора:

$q = (s, \vec{v})$, $s = a$, $\vec{v} = (b, c, d)$, множення кватерніонів буде виражатись через скалярний та векторний добутки 3-вимірних векторів:

$$q_1 q_2 = (s_1, \vec{v}_1)(s_2, \vec{v}_2) = (s_1 s_2 - \vec{v}_1 \cdot \vec{v}_2, s_1 \vec{v}_2 + s_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2).$$

Матриця повороту для кватерніонів

Поворотів $\vec{v}$ на кут α відносно осі $\vec{u}$ (одичний вектор) за допомогою одичного кватерніона $q = a + bi + cj + dk$

$$q = \left(\cos \frac{\alpha}{2}, \vec{u} \sin \frac{\alpha}{2} \right), \quad \vec{u} = (x, y, z);$$

відповідає наступна матриця повороту

$$R = \begin{pmatrix} 1 - 2(c^2 + d^2) & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & 1 - 2(b^2 + d^2) & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & 1 - 2(b^2 + c^2) \end{pmatrix}.$$

Матриця повороту для кутів Ейлера

Будь-яку просторову орієнтацію можна задати поєднанням трьох елементарних обертань, починаючи від відомої стандартної орієнтації.

Так само, будь-яка матриця повороту R може бути розкладена у вигляді **добутку трьох елементарних матриць повороту**.

Матриця повороту виражається через кути Ейлера так:

$$R = R_z(\psi) \cdot R_x(\theta) \cdot R_z(\varphi) =$$

$$\begin{pmatrix} \cos \psi \cos \varphi - \sin \psi \sin \varphi \cos \theta & -\sin \psi \cos \varphi \cos \theta - \cos \psi \sin \varphi & \sin \psi \sin \theta \\ \sin \psi \cos \varphi + \cos \psi \cos \theta \sin \varphi & \cos \psi \cos \theta \cos \varphi - \sin \psi \sin \varphi & -\cos \psi \sin \theta \\ \sin \theta \sin \varphi & \cos \varphi \sin \theta & \cos \theta \end{pmatrix}$$

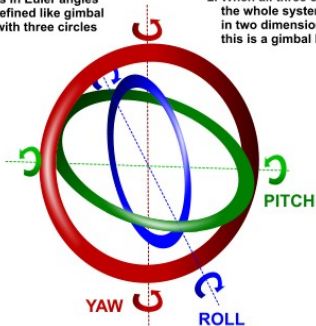
Переваги кватерніонів у визначенні поворотів

Представлення повороту в тривимірному просторі у вигляді одиничних кватерніонів в порівнянні з кутами Ейлера має деякі переваги:

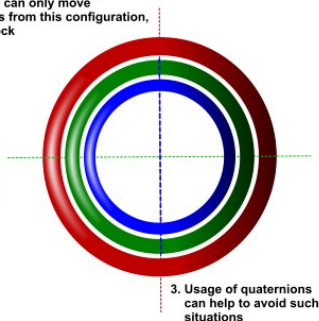
- Конкатенація обертань обчислюється швидше і чисельно більш стабільна.
- Виділення кута повороту і осі обертання є простішим.
- Інтерполяція є більш простою. (slerp - лінійна інтерполяція на сфері)
- Кватерніони не мають проблеми блокування обертання, на відміну від кутів Ейлера.

Карданнів підвіс, Gimble lock

1. Rotations in Euler angles can be defined like gimbal system with three circles



2. When all three circles are lined up, the whole system can only move in two dimensions from this configuration, this is a gimbal lock



3. Usage of quaternions can help to avoid such situations

Отримання кутів

Процес роботи з платою:

Калібрування:

- становлення кутів

$$\delta\phi^2 + \delta\psi^2 + \delta\varphi^2 < 0.01 ;$$

- знаходження g;

Робочий цикл:

- знаходження кутів
- знаходження прискорень в системі відліку плати
- поворот прискорень в лабораторну систему відліку

- віднімання g від a_z

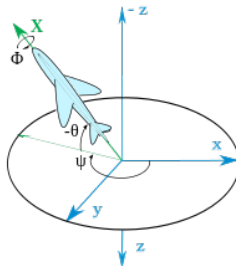


Figure 1 :

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix}$$

Фільтр Мадвіга

- виправлення помилок
- Profit!



Задача

Основною задачею є вирахування швидкостей.

Ми спробували інтегрування з фільтром низьких частот, але бажаного результату це не дало.

■ ???

Дякую за увагу!